On Maximizing Sensor Network Lifetime by Energy Balancing

Rong Du*, Lazaros Gkatzikis*, Carlo Fischione*, Ming Xiao†
* Automatic Control Department, † Communication Theory Department
KTH Royal Institute of Technology, Stockholm, Sweden
Email: {rongd, lazarosg, carlofi, mingx}@kth.se

***Abstract*—Many physical systems, such as water/electricity distribution networks, are monitored by battery-powered Wireless Sensor Networks (WSNs). Since battery replacement of sensor nodes is generally difficult, long-term monitoring can be only achieved if the operation of the WSN nodes contributes to a long WSN lifetime. Two prominent techniques to long WSN lifetime are i) optimal sensor activation and ii) efficient data gathering and forwarding based on compressive sensing. These techniques are feasible only if the activated sensor nodes establish a connected communication network (connectivity constraint), and satisfy a compressive sensing decoding constraint (cardinality constraint). These two constraints make the problem of maximizing network lifetime via sensor node activation and compressive sensing NP-hard. To overcome this difficulty, an alternative approach that iteratively solves energy balancing problems is proposed. However, understanding whether maximizing network lifetime and energy balancing problems are aligned objectives is a fundamental open issue. The analysis reveals that the two optimization problems give different solutions, but the difference between the lifetime achieved by the energy balancing approach and the maximum lifetime is small when the initial energy at sensor nodes is significantly larger than the energy consumed for a single transmission. The lifetime achieved by the energy balancing is asymptotically optimal, and that the achievable network lifetime is at least 50% of the optimum. Analysis and numerical simulations quantify the efficiency of the proposed energy balancing approach.**

***Index Terms*—network lifetime, energy balancing, sensor network, cyber-physical system**

I. Introduction

Wireless sensor networks (WSNs) are being used to monitor critical infrastructures in smart cities, such as water distribution networks, tunnels, bridges, and towers. Since sensor nodes are generally power limited, and battery replacement is difficult or even impossible, network lifetime is an important performance metric [1]. Several approaches have been proposed to prolong network lifetime and hence to enable long-term monitoring. For example, sensor nodes can form clusters, where participating nodes take turn to act as cluster-head to balance the energy consumption of the nodes [2], [3], [4]. The nodes can optimize routing [5], [6] or use multi-hop short range communication [7] to save energy in the data transmission. Moreover, event-trigger mechanisms [8] can be used to reduce the transmitted data volume. The sensor nodes can also be put into sleep or idle mode to save energy [9], [10]. The methods to be used for energy saving should depend on the characteristics of the monitoring applications.

In this paper, we consider the case of using densely deployed sensors to monitor an area where node replacement is difficult. Such a dense sensor network has the following benefits:

- better detection of events;
- robustness to sensor failure and measurement errors because of the availability of redundant sensor nodes. Thus, the network operation is ensured even when some sensor nodes fail;
- reduced energy consumption in data transmission by exploiting multi-hop short range communication. Thus, network lifetime is increased.

Consequently, even though dense networks introduce a higher installation cost, they substantially reduce the maintenance cost in return, and, most importantly, may provide better monitoring performance.

We consider to use data compression in the data gathering process, together with a sleep/awake mechanism for the sensing process, to prolong lifetime for such a dense sensor network. A natural question is whether the usual approach of energy balancing, i.e., preferably use the nodes with more residual energy [11], would be a viable choice for maximizing network lifetime in this context.

As the sensor nodes are densely deployed, their measurements exhibit spatial correlations. Such correlations enable us to use compressive sensing (CS) to accurately estimate the state of the monitored infrastructure with a minimal number of measurements [12], [13]. Therefore, one may adopt a CS-based data gathering scheme, such that in every timeslot only a portion of sensor nodes is activated to sense and transmit data hop-by-hop to the sink nodes. It follows that the expected monitoring performance of such a system can be guaranteed by CS while its energy efficiency can be improved by turning off the rest of the sensor nodes.

In our previous works [14], [10], we proposed a CS based sensor activation scheme based on energy balancing for dense WSNs to monitor water distribution networks. The devised data gathering scheme activates only a few connected sensor nodes for sensing and data transmission, to reduce the overall energy consumption and so that the monitoring performance is guaranteed even under sensor failures. However, whether that algorithm (or any energy-balancing based one) can achieve the maximum WSN lifetime is an open issue that has not

This work is supported by the Wireless@KTH Seed Project LTE-based Water Monitoring Networks and ICT EIT Lab project I3C.

been investigated before in the CS context. In summary, we address this fundamental issue of whether, in the considered WSN scenario, the energy balancing problem is equivalent to lifetime maximization. The main results of the paper are as follows:

- In order to provide insight on the complexity and the structure of the Lifetime maximization Problem, we cast it as a Multi-dimensional Knapsack problem , for which a rich literature on solution approaches exists.
- We provide an easy to calculate upper bound of maximum lifetime based on a transformation to a maximum flow problem, as shown by Theorem 1 in Section IV A.
- We propose an algorithm that gives an approximate solution to the maximum lifetime problem. The algorithm is based on the solution to an energy balancing problem. We show that such an algorithm is asymptotically optimal (as given by Theorem 2 in Section IV C) and the worst case approximation ratio (the ratio of the lifetime achieved by the algorithm to the optimal lifetime) is 50% (as shown in Theorem 3 in Section IV C). The asymptotic optimality and the approximation ratio of the algorithm constitute major original contributions of this paper because for maximum lifetime problems with connectivity and cardinality constraints there are no known optimality bounds.

The organization of the rest of the paper is as follows. We provide an overview of related literature in Section II. The formulation of the lifetime maximization problem and the energy balancing problem are described in Section III. In Section IV, the performance of the proposed algorithm in terms of network lifetime is analyzed. Numerical evaluations are provided in Section V. The conclusions of this work are presented in Section VI.

II. Related Work

A. Lifetime maximization by flow approximation

The lifetime of a network greatly depends on the residual energy of the participating nodes. There are different models for the energy consumption of sensor nodes. In [15], the energy consumption is linearly related to the receiving power, transmitting power and data transmission rate, and the expected lifetime of a sensor node is defined as the ratio of the energy capacity of the node and the average energy consumption. In [16], a slice model is used where the monitored area is partitioned into slices, each of which contains all the nodes that have the same hop distance from the sink node. In this model, the energy consumption of the nodes depends also on the distance (hop count). However, the distances of a node to any node in the same slice are considered to be the same. In [9], the energy consumptions of the nodes in sleep mode are assumed to be 0, whereas that of the active nodes follows an independent and identical distribution. In [17], a non-linear dynamic energy consumption model, called kinetic battery model, is used. In [18], given the fixed topology of a WSN, the energy consumption rate of the working sensor nodes is considered constant, whereas the consumption of the sleeping sensor nodes is 0. Similar to [18], we normalize the energy consumption of the active sensor nodes to 1 and set that of the nodes in sleep mode equal to 0.

One common way to prolong network lifetime is by reducing energy consumptions of each node. For example, in the event-trigger based approaches [8], one could reduce the sampling rate of the sensors, to save energy of sensing and data transmission. Notice that for general networks, the data transmission constitutes the major part of a node's energy consumption. Thus, several solutions have been considered to reduce nodal energy consumption due to data transmission, such as controlling the transmission power [19], [20] and compressing the measurements [21], [22] to be transmitted. Besides, harvesting energy from ambient environment [23], [24], [18], such as solar energy and vibration, or transmitting energy to the nodes wirelessly [25], [26], can also extend the lifetime of a node, but are beyond the scope of this work.

Besides the nodal perspective, prolonging lifetime from the network perspective, e.g., by optimized routing, has been widely studied. In this context, the classic flow approximation is commonly used [15], [16]. In particular, the energy budget of each node is represented as the number of flows that can pass the node, which is referred to as 'vertex capacity'. Then, finding the route in each wireless communication timeslot to maximize network lifetime is equivalent to finding the maximum flow from source node to sink node. In the seminal work [15], the energy consumption of the network has been modelled as a function of the traffic flow routing decisions. Then the problem is cast as a linear programming problem. In a similar network setting, where every sensor node can either transmit its data to its neighbor with low energy cost, or transmit data directly to the sink node with high energy cost, maximizing network lifetime is equivalent to flow maximization and energy balancing [16]. In such scenario, energy balancing has been used to maximize network lifetime [27], [28]. Another way to balance the energy consumption is rotating the working period of sensor nodes, i.e., allowing some sensor nodes to sleep without sacrificing in the monitoring performance. For instance, Misra et. al. [29] have considered finding different connected dominating sets of the WSN to prolong network lifetime. In each timeslot, only the sensor nodes in the connected dominating set are active and the other nodes are put into sleep. To rotate the working period of the nodes, it is desired to find the maximum connected domatic partition, which divides the WSN into as many as possible disjoint connected dominating sets. A similar problem has been considered in [18], where the sensor nodes have the energy harvesting ability. Thus, the working schedules of the connected dominating sets are also taken into consideration.

Compared to the WSN scenarios mentioned above, CS considered in this paper introduces a cardinality constraint. It is an open question how the problem of lifetime maximization of such WSN is related to the energy balancing problem. Another major difference of our study is that, in most of previous studies, the network lifetime depends on the minimal nodal lifetime, i.e., the network depletes once a node in the network depletes, since the sensor nodes in those scenarios monitor different events; here, due to the correlations of

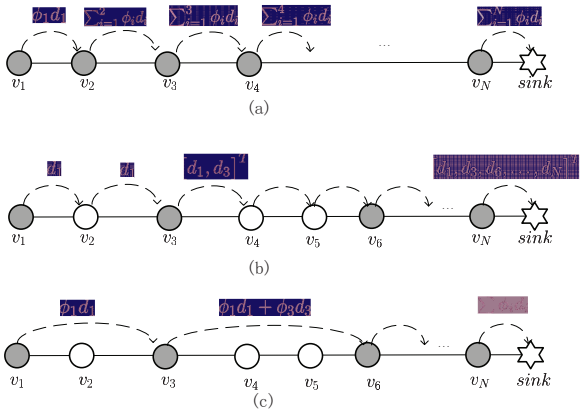


Fig. 1. Data gathering in (a) Compressive Data Gathering (CDG) [30]; (b) Compressed Sparse Function (CSF) [13]; (c) Proposed scheme, where d_i is the measurement of sensor node v_i, and $\boldsymbol{\phi}_i$ is a sampling vector with size $M \ll N$. In (a) and (c), every active node multiplies its measurement d_i with vector $\boldsymbol{\phi}_i$, adds the result with the vector received from the previous active node, and then transmits the new vector to its next hop node. Because the active nodes transmit vectors with size M, they have the same size of payloads.

measurements, CS enables us to care only about the number of activated sensor nodes. The network is considered depleted only when either there are not enough remaining sensor nodes to satisfy the cardinality requirement, or the remaining sensor nodes, with the sink nodes, can not form a connected graph. Thus, the flow approximation and the existing solutions cannot be directly applied in our setting. In this paper we adopt the concept of energy balancing and we characterize its relation to the maximum lifetime.

B. Compressive sensing for data gathering

For the sake of completeness, we describe the operation of existing data gathering schemes based on CS. Suppose that the sink node in a network wants to collect the measurements of N sensor nodes. In work [30], the proposed compressive data gathering (CDG) algorithm operates as shown in Fig. 1 (a), in which each sensor node multiplies its local measurement $d_i \in \mathbb{R}$, for $i = 1, \ldots, N$ with a random vector $\boldsymbol{\phi}_i$ of dimension $M \ll N$, adds the product $\boldsymbol{\phi}_i d_i$ with the vector $\sum_{k=1}^{i-1} \boldsymbol{\phi}_i d_i$ it receives from its neighbour, and then transmits the summation $\sum_{k=1}^{i} \boldsymbol{\phi}_i d_i$ to its next-hop sensor node. At the sink node, the measurements of the sensor nodes, $\boldsymbol{y} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]\boldsymbol{d}$, are recovered by solving an l_1-minimization problem as follow:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_{l_1} \quad \text{s.t. } \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}\|_{l_2}^2 \leq \varepsilon\,,$$

where $\boldsymbol{\Psi}$ is the basis on which $\boldsymbol{d}$ is sparse, and where l_1 and l_2 are the Manhattan norm and the Euclidean norm respectively. Then, $\hat{\boldsymbol{d}} = \boldsymbol{\Psi}\hat{\boldsymbol{x}}$. In this case, every sensor node only transmits messages of size $O(M)$, which balances the energy consumptions of the sensor nodes and also reduces the overall data traffic.

Fig. 1 (b) shows the Compressed Sparse Function (CSF) algorithm [13] for data gathering. In the figure, the grey sensor nodes sense and transmit their measurements to the next-hop sensor nodes, whereas the white ones act as relay nodes. As long as the sink node collects M out of N data measurements, the CSF algorithm can recover the remaining $N - M$ measurements. The idea is based on the mapping of the reading of the sensor nodes to their locations (or ids). Denote the mapping function $f(x)$ where x is the location of the id of the sensor nodes. Since the sensor nodes are densely deployed, $f(x)$ could be represented as $f(x) = \boldsymbol{\Psi}(x)^T\boldsymbol{c}$, where $\boldsymbol{\Psi}(x)^T$ is a basis such as the type-IV DCT basis, and $\boldsymbol{c}$ is the sparse coefficient of $f(x)$ on the basis. Then, suppose the ids of the activated sensor nodes are $a_1, \ldots, a_M$, the sink node has the data of $\boldsymbol{y} = [f(a_1), \ldots, f(a_M)]^T = [\boldsymbol{\Psi}(a_1)^T, \ldots, \boldsymbol{\Psi}(a_M)]^T\boldsymbol{c}$. The sink node estimates $\boldsymbol{c}$ by the l_1-minimization problem similar to the CDG approach, and then the mapping function $f(x)$ is retrieved. Based on $f(x)$, the measurements of the white nodes are also retrieved.

Considering a dense network, in our previous work [14], we proposed the data gathering scheme shown in Fig. 1 (c), where only the grey nodes are active and transmit data in the CDG way to the sink node. Since each active node transmits a calculated vector based on the summation of its measurement and its received vector, the packet sizes of the nodes are the same, and the energy consumption of the active nodes is balanced. The sink node first uses l_1-minimization to estimate the measurement of the grey sensor nodes, and then we use CSF to estimate the measurement of the white sensor nodes. The overall data traffic is further reduced, and the energy consumption of the grey sensor nodes is balanced. If the activation of the sensor nodes is decided carefully in each monitoring timeslot, the energy consumption of all the sensor nodes can be approximately balanced. However, understanding whether the network lifetime can be maximized by such an energy balancing approach is a fundamental open question.

Compared to our previous work [10], the major difference of this paper is two-fold: 1) our previous work provides an efficient method to solve the energy balancing problem, whereas this paper characterizes the performance of energy balancing in terms of network lifetime; 2) in this paper, we achieve another upper bound of network lifetime by transforming the original problem into a maximum flow with cardinality constraint problem. This upper bound is tighter than the bound of [10] and provides useful insight for the structure of the problem.

III. Problem Formulation and Preliminaries

We consider a WSN consisting of two tiers of nodes that monitors an area of line shape. Characteristic such examples are a pipeline in a water distribution network, a tunnel, or a bridge. The first tier consists of battery-powered sensor nodes that are densely deployed in the monitored area. Their role is to sense and relay data to a set of sink nodes. The second tier consists of sink nodes, which are grid-powered and are deployed at the two ends of the line. They collect data from the sensor nodes, and transmit the data to a remote monitoring center. Due to the length of the monitored area and the comparative small communication range of the sensor nodes, a multi-hop communication path from the sensor nodes to the sink node has to be established.

Since battery replacement is not easy for the applications mentioned above, a main objective is to maximize the network

TABLE I
MAJOR NOTATIONS USED IN THE PAPER

symbols	meanings
$\mathcal{G}(\boldsymbol{x}(t))$	induced graph of active sensor nodes and sink nodes
$\mathcal{N}(v_i)$	set of neighbour nodes of v_i
C_{ij}	capacity of arc $\langle v_i^{\text{out}}, v_j^{\text{in}} \rangle$
E_i	ratio of nodal battery to nodal energy consumption of v_i
M_c	minimum number of sensor nodes to be activated to satisfy connectivity constraint
M_{cs}	required number of active sensor nodes in a timeslot due to compressive sensing
N	number of sensor nodes
$\bar{T}^f$	upper bound of lifetime achieved by maximum flow approach
T_G	lifetime achieved by energy balancing approach
$T_{\max}$	maximum lifetime of the network
$\boldsymbol{Q}_i$	a column vector representing a feasible activation profile
$\boldsymbol{R}_i$	a route from s_l to s_r, and is represented by a list of nodes in the route
$\boldsymbol{R}_i^+$	set of forward arcs in $\boldsymbol{R}$
$\boldsymbol{R}_i^-$	set of backward arcs in $\boldsymbol{R}_i$
p_i	normalized residual energy of v_i
s_l and v_0	the leftmost sink
s_r and v_{N+1}	the right most sink
$u_{i,j,t}$	flow from v_i to v_j at timeslot t
v_i	sensor node i
$x_i(t)$	activation of v_i at timeslot t
z_i	number of activations of $\boldsymbol{Q}_i$

lifetime. Intuitively, it is beneficial to keep alive as much of the sensor nodes as possible, which motivates the design of activation algorithms based on energy balancing, i.e., preferably activate the nodes with more residual energy. Thus, the major problem to be considered here, is whether the maximum network lifetime can be achieved by the energy balancing approach, or (if not), what is the performance of the energy balancing approach in terms of network lifetime.

In the following, we use a scenario of pipeline monitoring for water distribution network to better illustrate the necessary concepts. However, all the provided results hold for any linear network. The major notations are listed in Table I.

A WSN for monitoring a single pipeline can be represented by a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertex set $\mathcal{V}$ represents both the sensor nodes and the sink nodes, and edge set $\mathcal{E}$ represents the links among nodes. Suppose there are N sensor nodes, then we denote s_l the leftmost sink node, s_r the rightmost sink node, and $v_1, v_2, \ldots, v_N$ the sensor nodes from left to right. For simplicity, sink nodes s_l and s_r are also represented as v_0 and v_{N+1}. Let r_i be the transmission range of v_i, then $\langle v_i, v_j \rangle \in \mathcal{E}$ if and only if the distance between v_i and v_j is smaller than or equal to r_i. Also, we denote $\mathcal{N}(v_i) = \{v_j | \langle v_i, v_j \rangle \in \mathcal{E}\}$ the set of neighbours of v_i, and $\mathcal{N}_-(v_i) = \{v_j | v_j \in \mathcal{N}(v_i) \wedge j > i\}$ the downstream set of v_i. Our analysis relies on the following two assumptions that generally hold in water distribution networks[1]:

Assumption 1: All sensor nodes and the sink nodes are characterized by the same communication range $r_i = r$.

Remark 1: We suppose that the transmission power of the sensor nodes are fixed to a pre-set value. Since in a dense network, a node could have multiple neighbors for data relaying, the transmission power of a node could be set to the minimum value to save energy. Thus, the communication range of all the sensor nodes is the same. However, in the numerical simulations, we also examine the performance of Algorithm 1 when this assumption does not hold.

Assumption 2: All sensor nodes and sink nodes in the same pipeline are deployed in a line.

Remark 2: This assumption is reasonable as the diameter of the pipeline is small compared to the length of a pipeline. Thus, a pipeline sensor network can be considered as a line.

For energy saving purposes, nodes can be deactivated. Time is slotted. In a timeslot t, a sensor node can either be activated to sense and transmit data, or be in the sleeping mode. The activated sensor nodes transmit data in the CDG way [30], i.e., every node transmits a vector of the same size based on the projection of its measured data and the vector it receives. Thus, the payloads of the transmitted packets at each active node are the same, and the energy consumption is approximately the same. Therefore, we may normalize the energy consumption for the active sensor nodes in a timeslot to 1 for simplicity. Then the energy budget of v_i, $E_i \in \mathcal{Z}^+$, is characterized as the number of timeslots that a node can be activated. Let $E_i(t)$ denote the number of timeslots that v_i can be activated from timeslot t, which can be considered as the residual energy at t, and $E_i(1) = E_i$.

In the following, we first formulate the lifetime maximization problem. Then, we introduce an equivalent multi-dimensional knapsack problem, which allows us to solve the lifetime maximization problem for small networks. Last, we present the energy balancing problem and a solution algorithm, which are then used to study the achievable performance of energy balancing in terms of network lifetime.

A. Lifetime maximization problem

We define the lifetime of a WSN to be the operating time until either WSN becomes disconnected, or the monitoring performance of the WSN cannot be guaranteed. In each timeslot, the connectivity and the monitoring performance requirement of the active sensor network must be satisfied. Let binary variable $x_i(t)$ indicate whether node v_i is active at timeslot t. Then, the energy dynamics of v_i can be written as $E_i(t+1) = E_i(t) - x_i(t)$, and the scheduling problem considered in this paper is to determine $\boldsymbol{x}(t) = [x_1(t), \ldots, x_N(t)]^T, \forall t$.

Let $\mathcal{G}(\boldsymbol{x}(t))$ denote the induced graph of active sensor nodes and the sink nodes. Then, the connectivity constraint is defined as follows:

[1]Those assumptions are necessary to derive the analytical results. More details on the general performance can be found in our previous work [10].

Definition 1: **(Connectivity Constraint)** The activation of the sensor nodes $\boldsymbol{x}(t)$ satisfies the connectivity constraint if and only if the induced graph $\mathcal{G}(\boldsymbol{x}(t))$ is connected.

Remark 3: To check the connectivity of a subgraph, one approach could be breadth-first search. However, since the connectivity checking is not the major objective of the paper, we do not elaborate further. More details can be found in Chapter 3.2 of [31].

Regarding monitoring performance, our previous works [14], [10] have shown that the estimation error by CS is related to the number of active nodes, m, where m is much smaller than the number of sensor nodes N, i.e., $m \ll N$. Thus, the requirement on monitoring performance can be specified as a cardinality constraint defined as follows:

Definition 2: **(Cardinality Constraint)** The activation of the sensor nodes $\boldsymbol{x}(t)$ satisfies the cardinality constraint if and only if $\sum x_i(t) \geq M_{\text{cs}}$, where M_{cs} is determined by the required estimation error of the measured data.

Then, the lifetime maximization problem can be formulated as an optimal control problem as follows:

$$\max_{\boldsymbol{x}(t), t=1,\ldots,T} \quad \sum_{t=1}^{T} 1 \tag{1a}$$

$$\text{s.t.} \quad E_i(t+1) = E_i(t) - x_i(t), \forall i, 1 \leq t \leq T, \tag{1b}$$

$$E_i(1) = E_i, \forall i, \tag{1c}$$

$$E_i(t) \geq 0, \forall i, 1 \leq t \leq T+1, \tag{1d}$$

$$\mathcal{G}(\boldsymbol{x}(t)) \text{ is connected }, \forall t, \tag{1e}$$

$$\sum_{v_i \in \mathcal{V}} x_i(t) \geq M_{\text{cs}}, \forall 1 \leq t \leq T, \tag{1f}$$

$$x_i(t) \in \{0,1\}, \forall i, t\,, \tag{1g}$$

where Constraint (1b) is the dynamic of the energy of the sensor nodes, Constraint (1c) is the initial state of the WSN, (1d) is the non-negative constraint on the energy of sensor nodes, (1e) is the connectivity constraint, (1f) is the cardinality constraint. Also, we have that Constraints (1b)-(1d) can be equivalently captured by the following energy budget constraint:

$$\sum_{t=1}^{T} x_i(t) \leq E_i, \forall i.\,, \tag{2}$$

where the proof is in Appendix A.

Notice that, given a feasible $\boldsymbol{x}(t), \forall t$ that satisfies Constraints (2), and (1g), we can construct $E_i(1) = E_i$, and $E_i(t+1) = E_i(t) - x_i(t)$, such that $E_i(t) \geq 0, \forall i, 1 \leq t \leq T+1$ always holds. Thus, indeed we can replace Constraints (1b)-(1d) by Constraint (2). This optimization problem is particularly challenging due to the binary nature of activation variables and cardinality constraint (1f), as we articulate below.

Remark 4: Note that Problem (1) addresses only which set of nodes should be activated, and not how the routing of measurements to the sink in a multi-hop fashion should be performed. Under Assumptions 1 and 2, all vertices in a connected subgraph of $\mathcal{G}$ can form a connected routing path, i.e., they are in a line topology and every node receives data from at most one node and transmits data to at most one node (see more details from the proof Theorem 1 in our previous work [10]). Thus, it is guaranteed that a forward to the nearest active neighbour routing is always an optimal routing strategy. In other words, the induced graph of the active sensor nodes and the sinks is a connected graph, if and only if there is a path in the induced graph from v_0 to v_{N+1} that passes only through all the active nodes. Thus, it is equivalent to replace Constraint (1e) by the requirement of the existence of path from v_0 to v_{N+1} for any t without changing the optimal solution of the original problem, as we will do in Section IV to simplify analysis.

B. Knapsack approximation for small WSNs

Lifetime maximization Problem (1) is NP-Hard [10]. In order to provide insight on the complexity and the structure of the problem, we show that it can be cast as a knapsack optimization problem, for which a solution method is known. However, the method is practical only for small networks. We also use it in the numerical evaluation part as a benchmark.

To begin with, we define activation profile as follow:

Definition 3: **(Activation Profile)** An activation profile is a group of sensor nodes that satisfies the connectivity constraint. We say an activation profile is feasible if and only if it also satisfies the cardinality constraint.

Then, we may equivalently reformulate the maximum lifetime problem as stated in the following lemma:

Lemma 1: Consider a WSN $\mathcal{G}$. Let L be the total number of feasible activation profiles. Each feasible activation profile is represented by a column vector $\boldsymbol{Q}_l = [q_l(1), \ldots, q_l(N)]^T$. Here, $q_l(i) = 1$ if and only if sensor node v_i belongs to profile l, otherwise $q_l(i) = 0$. Define vector $\boldsymbol{z} = [z_1, \ldots, z_L]^T$, where z_i denotes the number of timeslots that profile i is chosen for activation. Then, the lifetime maximization Problem (1) is equivalent to the following problem:

$$T = \max_{\boldsymbol{z}} \quad \sum_{l=1}^{L} z_l \tag{3a}$$

$$\text{s.t.} \quad \boldsymbol{Q}\boldsymbol{z} \leq \boldsymbol{E}, \tag{3b}$$

$$z_l \in \mathcal{Z}^+, \forall l \in \{1, \ldots, L\}\,, \tag{3c}$$

where $\boldsymbol{Q} = [\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_L]$ is the set of all feasible profiles, $\boldsymbol{E} = [E_1, \ldots, E_N]^T$ is the vector of initial energies of all sensor nodes, and $\mathcal{Z}^+$ is the set of non-negative integers.

Proof: The proof is in Appendix B. ■

Lemma 1 shows that the maximum lifetime problem under cardinality and connectivity constraints can be turned into a multi-dimensional knapsack (MDK) problem [32]. In our case, the knapsack corresponds to the energy budget of each sensor nodes, and each profile corresponds to an item of value 1.

Remark 5: There are several methods to solve MDK problems, such as branch-and-bound, dynamic programming, and heuristic algorithms. We compare them in terms of complexity and optimality in Table II. Branch-and-bound, and dynamic programming can achieve optimal solution, however, the complexity of branch-and-bound is as high as exhaustive search in the worst case, whereas dynamic

TABLE II
COMPARISON OF DIFFERENT METHODS FOR MULTI-DIMENSIONAL KNAPSACK PROBLEM

Methods	Complexity	Optimality
Branch-and-Bound	Unknown	Yes
Dynamic Programming	Curse of dimension	Yes
Heuristic	Low	Not sure

programming suffers from the curse of dimensionality and is not a scalable approach. Therefore, they cannot be applied in our scenario, where network size is large and nodes have limited computational and storage capability. However, we use branch-and-bound algorithm in simulation part for comparison purposes. Besides, heuristic methods such as genetic networks, even though with low complexity, cannot guarantee optimality. Thus, the transformation into an MDK problem is only valid for small scale network instances. For large scale (dense) networks, we consider approximating Problem (1) by an energy balancing problem described in the next subsection.

C. Energy balancing problem

In our previous work [14], we proposed an energy balancing problem, together with a solution method. Since in this paper we investigate the fundamental properties of the energy balancing problem from the point of view of network lifetime maximization, we give the necessary details in the following:

Recall that $E_i(t)$ is the residual energy of v_i at timeslot t, we define its normalized residual energy as $p_i(t) = E_i(t)/E_i$. We denote $\mathcal{V}(t) = \{v_i \in \mathcal{V} | E_i(t) \geq 1\}$ the set of candidate sensor nodes to be potentially activated at timeslot t. Then, we posed the energy balancing problem as a sequence of problems, and one for each timeslot t as follows:

$$\max_{\boldsymbol{x}} \sum_{i \in \mathcal{V}} x_i p_i \tag{4a}$$

$$\text{s.t.} \quad \sum_{v_i \in \mathcal{V}} x_i = \max\{M_{\text{cs}}, M_c\}, \tag{4b}$$

$$\mathcal{G}(\boldsymbol{x}) \text{ is connected}, \tag{4c}$$

$$x_i \in \{0,1\}, \ \forall i \in \mathcal{V}, \tag{4d}$$

where time index t is discarded for notational simplicity, and M_c is the minimum number of sensor nodes that must be activated to satisfy the connectivity constraint.

The intuition behind the energy balancing approach is that, in each timeslot, the number of active sensor nodes has to be as small as possible. Out of the feasible activation profiles with the same number of active sensor nodes, the profile with maximum normalized residual energy is preferred. Recall that M_{cs} is the minimum number of sensor nodes that should be activated to ensure monitoring performance. When $M_{\text{cs}} < M_c$, one cannot find a route from s_l to s_r which passes exactly M_{cs} sensor nodes. In this case, the requirement of $\sum_{v_i \in \mathcal{V}} x_i = M_{\text{cs}}$ contradicts to Constraint (4c). It gives us that $\sum_{v_i \in \mathcal{V}} x_i$ should be M_c instead. Thus, even though Constraint (4c) is given, the M_c in Constraint (4b) is not redundant.

In [14], we developed an algorithm to solve Problem (4). The details of the procedure are shown in Algorithm 1. First, Algorithm 1 finds M_c by a shortest path algorithm such as Dijkstra's algorithm in Line 1, namely finds the shortest path from v_0 to v_{N+1}, where the weights of all the edges are 1. Then, the minimum number of sensor nodes, m, that satisfies both the connectivity and the cardinality constraints is calculated in Line 3. Knowing m, we can solve Problem (4) by dynamic programming (Line 4), where $g(v_i, k)$ represents the maximum sum of normalized residual energy of k connected sensor nodes among v_{i+1} to v_N, and it is calculated as

$$g(v_i,k) = \begin{cases} \max\limits_{v_j \in \mathcal{N}'_-(v_i)} \{g(v_j, k-1) + p_j\} & \text{if } k>0, \mathcal{N}'_-(v_i) \neq \emptyset \\ 0 & \text{if } k=0, s_r \in \mathcal{N}_-(v_i) \\ -\infty & \text{otherwise}, \end{cases} \tag{5}$$

where p_j is the normalized residual energy of sensor node v_j. Recall that $\mathcal{N}_-(v_i) = \{v_j | v_j \in \mathcal{N}(v_i) \wedge j > i\}$ is the downstream set of v_i, $\mathcal{N}'_-(v_i) = \mathcal{N}_-(v_i) \backslash \{s_r\}$ is the set of sensor nodes in the downstream set of v_i. Notice that $g(v_i, k)$ is directly related to $g(v_j, k-1)$, where v_j is in the neighbour of v_i. Thus, the nodes determined by this dynamic programming are connected.

In our previous work [10], we proposed to activate the sensor nodes as suggested by the solution of the energy balancing problem (4) in each timeslot. Then update the nodal normalized residual energy to be the input of the energy balancing problem in the next timeslot, until the problem is

Algorithm 1 Optimal activation schedule for Problem (4) [10]

Input: Adjacency matrix $\boldsymbol{A}$, the minimum number of active node M_{cs}, and the normalized residual energy of the sensor nodes $\boldsymbol{p}$.
Output: A set of sensor nodes $\mathcal{V}_A$ that need to be activated.
1: Find the minimum number of sensor nodes, M_c, that satisfy the connectivity constraint
2: **if** $M_c < +\infty$ **then**
   // Find the minimum number of nodes that satisfy both connectivity constraint and cardinality constraint
3: $m = \max\{M_c, M_{\text{cs}}\}$
   // Solve Problem (4) by dynamic programming
4: Calculate $g(s_l, m)$ according to Eq. (5) and the corresponding $\mathcal{V}_A$
5: **return** $\mathcal{V}_A$
6: **else**
7: **return** $\emptyset$
8: **end if**

Algorithm 2 Activation based on Energy Balancing [10]

Input: Adjacency matrix $\boldsymbol{A}$, the minimum number of active node M_{cs}, the battery of the nodes $E_i, \forall i$
Output: Network lifetime T
1: Set $t \leftarrow 1$, $Flag \leftarrow TRUE$, $E_i(t) = E_i$
2: **while** $Flag$ **do**
3: Set $p_i \leftarrow E_i(t)/E_i$, $\boldsymbol{p} = \{p_1, \ldots, p_N\}$
   // Node activation based on Algorithm 1
4: Find M_c for the connectivity constraint
5: $\mathcal{V}_A \leftarrow$ Call Algorithm 1 with input $\boldsymbol{A}, M_{cs}, \boldsymbol{p}$
6: **if** $\mathcal{V}_A \neq \emptyset$ **then**
7: Set $t \leftarrow t+1$, $E_j(t) \leftarrow E_j(t-1) - 1, \forall j \in \mathcal{V}_A$
8: **else**
9: Set $Flag \leftarrow FALSE$
10: **end if**
11: **end while**
12: **return** $T \leftarrow t-1$

infeasible, as described by Algorithm 2. However, whether this approach could lead to the maximum network lifetime has not been analyzed before. Thus, *the investigation of the fundamental properties of energy balancing in terms of network lifetime with a cardinality constraint* is the core contribution of this paper.

IV. Performance Analysis of the Optimal Activation Schedule

In this section, we analyze the performance of Algorithm 2 in terms of network lifetime. We transform the lifetime maximization Problem (1) to a maximum flow problem with a typical vertex capacity constraints and a new cardinality constraint. Such transformation enables a better understanding of the maximum lifetime problem, provides us a new lifetime upper bound, and also enables us to derive the performance bound of the proposed Algorithm 2.

A. Lifetime maximization as a maximum flow problem

The maximum lifetime Problem (1) is a maximum flow problem with vertex capacities (see [33] for a description of these problems) with an additional cardinality constraint. Let $u_{i,j,t}$ denote the flow from node v_i to v_j in timeslot t. Then, the maximum flow with vertex capacity and cardinality constraints is formulated as follows:

$$\max_{\boldsymbol{u}} \quad T \tag{6a}$$

$$\text{s.t.} \quad \sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t} - \sum_{v_j : v_i \in \mathcal{N}(v_j)} u_{j,i,t} = \begin{cases} 1, & i = 0, \forall t = 1, \ldots, T \\ -1, & i = N+1, \forall t = 1, \ldots, T \\ 0, & \forall 1 \le i \le N, \forall t = 1, \ldots, T \end{cases}, \tag{6b}$$

$$\sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t} \le 1, \forall i, t \tag{6c}$$

$$\sum_{t=1}^{T} \sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t} \le E_i, \forall i = 1, \ldots, N, \tag{6d}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t} \ge M_{\text{cs}} + 1, \forall t = 1, \ldots, T, \tag{6e}$$

$$u_{i,j,t} \in \{0, 1\}, \forall i, j, t\,, \tag{6f}$$

Then, we have the following lemma:

Lemma 2: Under Assumptions 1 and 2, Problem (1) is equivalent to Problem (6).

Proof: The proof is in Appendix C. ∎

Problem (6) is a binary programming problem. If Constraint (6f) is relaxed to $0 \le u_{i,j,k} \le 1, \forall i, j, k$, the problem becomes a linear programming problem. Then, an upper bound of WSN lifetime for Problem (1) can be established, as stated by the following theorem:

Theorem 1: Consider optimization Problem (1) for a WSN that follows Assumptions 1 and 2. The WSN lifetime is upper bounded by $\bar{T}^f$, where $\bar{T}^f$ is the optimal value of Problem (6) with Constraint (6f) relaxed as $0 \le u_{i,j,t} \le 1, \forall i, j, t$.

Proof: Suppose T^* is the optimal value of Problem (1), then it is also the optimal value of Problem (6) according to Lemma 2. Further, as $\bar{T}^f$ is the optimal value of the relaxed Problem (6), we have $T^* \le \bar{T}^f$ which completes the proof. ∎

Remark 6: This bound is quite tight if Assumptions 1 and 2 hold, as will be shown in Section V. Notice that the relaxed Problem (6) is a linear optimization problem and solvable. Consequently, we can use a bisection approach to find $\bar{T}^f$, as discussed in Appendix D. Besides, one may derive a good solution by rounding the result of the relaxed Problem (6)[2].

Based on Theorem 1, we study the performance of Algorithm 2 from the perspective of maximum flow problem. We first turn the maximum flow Problem (6) with vertex capacities to a maximum flow problem with edge capacities according to the following remark.

Remark 7: Problem (1) can be formulated as a maximum flow problem with edge capacities [34] and cardinality constraints. The basic idea is to substitute each node v_i with two nodes v_i^{in} and v_i^{out} connected by an arc $\langle v_i^{\text{in}}, v_i^{\text{out}} \rangle$ with capacity E_i. More details can be found in Appendix E.

Then, we show how the problem can be solved via a modified maximum flow algorithm. For such a purpose, we introduce some additional notations. Let f_{ii} be the flow on arc $\langle v_i^{\text{in}}, v_i^{\text{out}} \rangle$, and f_{ij} the flow on arc $\langle v_i^{\text{out}}, v_j^{\text{in}} \rangle$. The capacity of arc $\langle v_i^{\text{in}}, v_i^{\text{out}} \rangle$ is $C_{ii} = E_i$, and the capacity of arc $\langle v_i^{\text{out}}, v_j^{\text{in}} \rangle$ is $C_{ij} = +\infty$, as shown in Fig. 5. Given a route $\boldsymbol{R}_a = \langle s_l, v_{a_1}, \ldots, v_{a_k}, s_r \rangle$, we say that arc $\langle v_{a_i}, v_{a_{i+1}} \rangle$ belongs to the set of forward arcs of $\boldsymbol{R}_a$, which is denoted by $\boldsymbol{R}_a^+$, if and only if $\langle v_{a_i}, v_{a_{i+1}} \rangle \in \mathcal{E}'$. Otherwise, the arc belongs to the set of backward arcs of $\boldsymbol{R}_a$, which is denoted by $\boldsymbol{R}_a^-$. Then the maximum flow increment of $\boldsymbol{R}_a$ is defined as $\delta_a = \min\{\{C_{a_i a_j} - f_{a_i a_j} | \langle v_{a_i}, v_{a_j} \rangle \in \boldsymbol{R}_a^+\}, \{f_{a_i a_j} | \langle v_{a_i}, v_{a_j} \rangle \in \boldsymbol{R}_a^-\}\}$. $\boldsymbol{R}_a$ is said to be unblocked if and only if $\delta_a > 0$. Then we can use a modified Ford-Fulkerson Algorithm to find which nodes should be activated at each timeslot, so that the corresponding route at each timeslot is feasible.

B. A modified maximum flow algorithm based on Ford-Fulkerson Algorithm

The derived modified Ford-Fuklerson Algorithm works as follows. We find an unblocked route $\boldsymbol{R}_i$ from s_l to s_r in $\mathcal{G}'$ that contains no backward arcs and passes at least M_{cs} arcs with capacity less than $+\infty$. This is equivalent to passing at least M_{cs} sensor nodes in $\mathcal{G}$, and hence it corresponds to one route in $\boldsymbol{R}$. We perform an augmentation along route R_i with increment 1, i.e., the flows f of all the arcs in the route R_i increase by 1. Then, we find an unblocked route that contains no backward arcs until there is no such unblocked route in $\mathcal{G}'$ again. This operation gives a sequence of routes $\boldsymbol{R}(1), \boldsymbol{R}(2), \ldots, \boldsymbol{R}(T)$. If we are unable to find an unblocked route that contains backward arcs at $T+1$, then activation scheme $\boldsymbol{R} = [\boldsymbol{R}(1), \boldsymbol{R}(2), \ldots, \boldsymbol{R}(T)]$ leads to maximum network lifetime.

Notice that this algorithm does not allow choosing an unblocked route with backward arcs in each timeslot, it

[2]However, the main difficulty is to determine the rules of rounding such that the result satisfies both connectivity and cardinality constraints, and leads to the maximum network lifetime. This is left as a future work.

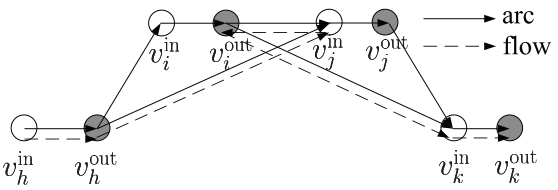


Fig. 2. A subnetwork that contains a backward arc $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$

requires an exhaustive search, and thus is not practical. However, it suggests the following lemma:

***Lemma** 3:* Consider a WSN satisfying Assumptions 1 and 2. If an unblocked route with backward arcs exists when Algorithm 2 terminates, we can alter one of the previous activation decisions to extend network lifetime by 1.

Proof: The proof is in Appendix F. ■

This lemma shows that *the existence of an unblocked route with backward arcs suggests the suboptimality of an activation*. Furthermore, extending the network lifetime in such way requires an unblocked route with backward arcs ($\boldsymbol{R}_b$ in the proof) and an unblocked route with no backward arcs ($\boldsymbol{R}_a$ in the proof) that we have selected before the WSN expires. This gives us the worst case approximation ratio of the lifetime achieved by Algorithm 2 to the maximum network lifetime, as will be shown in the next subsection.

C. Performance analysis of Algorithm 2

According to Lemma 3, the more the unblocked routes we can find, the more suboptimal the activation is. Thus, we can analyze the gap of lifetime we achieve by Algorithm 2 to the optimal lifetime value, by counting how many unblocked backward routes can be found when Algorithm 2 terminates.

Then, the performance of Algorithm 2 can be characterized by the following lemmas.

***Lemma** 4:* Consider a WSN $\mathcal{G}$ that satisfies Assumptions 1 and 2. Algorithm 1 is applied to determine the activation of the sensor nodes in each time slot. Let the achieved WSN lifetime be t_1, i.e., on t_1+1, no unblocked routes that contain only forward arcs can be found. If an unblocked route with backward arcs, $\boldsymbol{R}_b$, can be found at t_1+1, then $\boldsymbol{R}_b$ does not contain any backward arc $\langle v_i^{\text{out}}, v_i^{\text{in}} \rangle$ for any i.

Proof: The proof is in Appendix G. ■

According to Lemma 4, if one can find unblocked routes with backward arcs at the end, the backward arcs should be $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$, $j > i$. Thus, even though the route may contain several backward arcs, we can divide the route into several separated parts, each of which contains only one backward arc for easier analysis. Thus, we just need to focus on one of them as shown in Fig. 2.

***Lemma** 5:* Consider a WSN $\mathcal{G}$ that satisfies Assumptions 1 and 2. Algorithm 1 is applied to determine nodes to activate in each timeslot. If a backward arc $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$ $(j > i)$ exists in an unblocked route $\boldsymbol{R}_b$ when the WSN lifetime expires, then the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$ is 1.

Proof: The proof is in Appendix H. ■

From the proof of Lemma 5, we know that Algorithm 2 could be suboptimal due to the existence of unblocked routes with backward arcs. However, the maximum flow increment of the unblocked route with backward arcs does not increase when we multiply the $E_i, \forall i$ with a positive integer η. Then we have the following core result:

***Theorem** 2:* Consider a WSN that satisfies Assumptions 1 and 2, with initial energy $\boldsymbol{E}$. Let $T_{\max}(\eta)$ and $T_G(\eta)$ be the maximum network lifetime by Problem (1) and the network lifetime achieved by Algorithm 2, with initial energy $\eta\boldsymbol{E}$. Then

$$\lim_{\eta \to +\infty} \frac{T_{\max}(\eta) - T_G(\eta)}{T_{\max}(\eta)} = 0\,.$$

Proof: $T_{\max}(\eta) - T_G(\eta)$ is bounded by the number of unblocked routes with backward arcs when the network expires. According to Lemma 5, this number does not increase if the $E_i, \forall i$ are multiplied by a positive integer. Thus, $T_{\max}(\eta) - T_G(\eta)$ is bounded. However, we know that $T_{\max}(\eta) \geq \lfloor \eta \rfloor T_{\max}(1)$, and it tends to ∞ as η tends to ∞. Thus,

$$\lim_{\eta \to +\infty} \frac{T_{\max}(\eta) - T_G(\eta)}{T_{\max}(\eta)} = 0\,.$$

This concludes the proof. ■

Furthermore, based on Lemma 3, we have the approximation ratio of Algorithm 2 as shown in the following theorem.

***Theorem** 3:* Consider a WSN that satisfies Assumptions 1 and 2. Let $T_{\max}$ and T_G be the maximum network lifetime and the lifetime achieved by Algorithm 2. Then, we have $T_G \geq 0.5T_{\max}$.

Proof: According to Lemma 3, to extend the network lifetime requires us selecting an unblocked route with backward arcs and an unblocked route with no backward arcs before the WSN expires. We call such pair of routes an incremental pair. Furthermore, we know from Lemma 5 that the maximum flow increment of each backward arc is at most 1. Thus, when Algorithm 2 terminates, we have that, T_G, i.e., the summation of flows in unblocked routes without backward arcs, is larger or equal to the number of incremental pairs. Such number of incremental pairs is larger or equal to the additional network lifetime that can be extended by the backward arcs, which is $T_{\max} - T_G$. This gives us $T_G \geq 0.5T_{\max}$ and completes the proof. ■

***Remark** 8:* The provided approximation ratio is tight. An example for $T_G = 0.5T_{\max}$ can be shown using the topology of Fig. 2, where the left sink node is connected to v_h and v_i and the right sink node is connected to v_j and v_k, and the cardinality constraint requires us to pick 2 sensor nodes to activate in each time slot. The initial energy of v_i, v_j, v_h, v_k are all 1. Then, it is easy to achieve that the maximum network lifetime is 2, i.e., to activate v_i and v_k in one timeslot and to activate v_h and v_j in the other timeslot. In this case, the lifetime achieved by Algorithm 2 could be 1, if it picks v_i and v_j at the first timeslot. Then $T_G = 1 = 0.5T_{\max}$. However, Lemma 5 also gives us that, if the initial energy of all four sensor nodes are $E \gg 1$, the lifetime gap, $T_{\max}(E) - T_G(E)$, is always 1. In this case $T_{\max}(E) = 2E$ and $T_G(E) = 2E-1$. Thus, the gap is negligible when E is large enough, as suggested by Theorem 2. Therefore, when the nodal energy consumption in a timeslot is much smaller compared to the

TABLE III
GAP BETWEEN LIFETIME ACHIEVED BY ENERGY BALANCING (ALGORITHM 1) AND MAXIMUM LIFETIME (EXHAUSTIVE SEARCH)

Gaps (timeslots)	0	1	2	more
Number of cases	177	44	1	0

nodal battery capacity, which is generally true for sensor nodes for long term monitoring applications, the lifetime gap is negligible.

Theorem 2 shows that even though energy balancing is not equivalent to lifetime maximization in the considered network structure, the gap between the lifetime achieved by Algorithm 2 to the maximum network lifetime is small when the initial energies of the sensor nodes are large enough compared to the energy consumption in an active timeslot. It follows that Algorithm 2 can be used to derive good activation schedules for sensor nodes in terms of WSN lifetime.

For an illustration of the performance of Algorithm 2, numerical evaluations are given in Section V.

V. NUMERICAL EVALUATIONS

In this section, we evaluate numerically the lifetime achieved by energy balancing, and we compare it to the maximum lifetime. Suppose the length of the pipeline under study is L, we normalize it to be 1 for simplicity. Then the normalized transmission range of sensor nodes is r/L. Two sink nodes are deployed at the end point of the pipeline, one at point 0 and the other at point 1. The sensor nodes are uniformly randomly deployed in the range of $(0, 1)$. The initial energy of each node is randomly set according to a Gaussian distribution $E_i \sim \mathcal{N}(50, 5^2)$. In every timeslot, the energy of the active sensor nodes is reduced by 1, whereas the energy of other sensors remains the same. Once the residual energy of a sensor node is less than 1, it is considered as expired and is excluded from the available sensor node set $\mathcal{V}$. Once the sensor nodes in $\mathcal{V}$ become disconnected or their number is less than M_{cs}, the network has expired.

We first compare the network lifetime achieved by Algorithm 2 to the optimal solution by solving Problem (3) based on Branch-and-Bound method. As the number of nodes increases, the number of possible routes increases dramatically. Consequently, the size of the variables in the MDK Problem (3) also increases dramatically, and it becomes difficult to solve. Thus, we set the size of network to relatively small values, so that the MDK problem can be solved efficiently. The parameters of the network are as follows: the number of nodes, N, are randomly picked from 15 to 20, M_{cs} are randomly picked from 7 to 10, the normalized transmission range, r/L, is 0.25. We test 222 different random cases, among which there are 44 cases that the network lifetime by Algorithm 2 is 1 timeslot less than the optimal, and 1 case that is 2 timeslots less than the optimal, as shown in Table III. This supports our finding that balancing residual energy is effective to achieve the lifetime close to the maximum in the considered network.

We further compare the performance of Algorithm 2 to greedy based search with random (GBS+R) Algorithm and

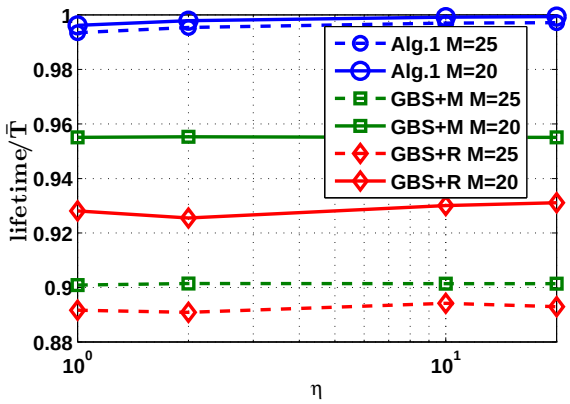
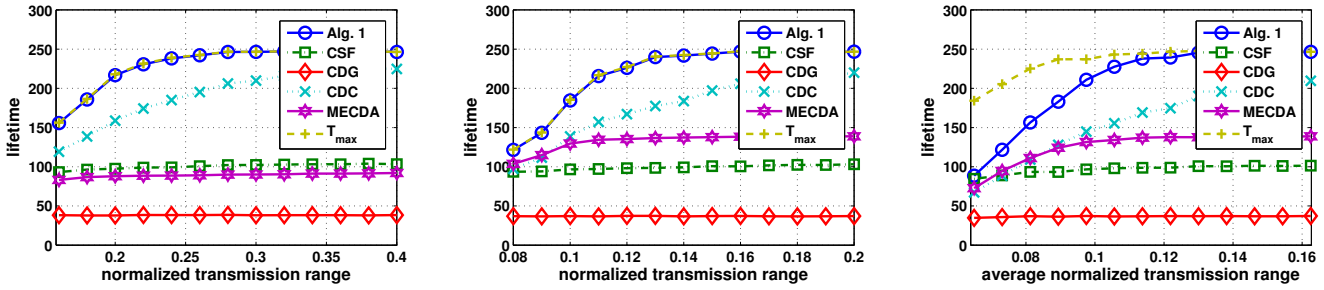


Fig. 3. Comparison of the ratio of network lifetime achieved by each algorithm to the upper bound of network lifetime with $N = 100, r/L = 0.15$.

greedy based search with maximum (GBS+M) Algorithm [14] as shown in Fig.3. We check the network lifetime when the initial energy of sensor nodes are multiplied by η. We calculate the network lifetime when η is $1, 2, 10, 20$, respectively, and then divide the lifetime by the upper bound of the network lifetime. For Algorithm 2, it is shown that the ratio of network lifetime achieved by the algorithm to the upper bound of network lifetime, $T_G(\eta)/\bar{T}(\eta)$, increases slightly as η increases. However, such a trend does not exist for the GBS+R and GBS+M Algorithm.

Last, we evaluate the performance of Algorithm 1 by comparing its performance to that of the state-of-art approaches, i.e., CDG [35], CSF [13], CDC [36], MECDA [37], and also by comparing the upper bound of WSN lifetime $\bar{T}^f$ achieved according to Theorem 1. The results for equal transmission range are shown in Fig. 4 (a) and (b). The horizontal axis represents the normalized transmission range, and the vertical axis is the average WSN lifetime. The WSN lifetime achieved by Algorithm 2 (blue solid line with circles) is very close to the upper bound (yellow dash line marked by plus) established by Theorem 1. It shows that the performance of Algorithm 2 is near optimal, and the upper bound by Theorem 1 is tight. The result of unequal transmission range is shown in Fig. 4 (c). In this case, the yellow dash line may not be the upper bound of the network lifetime because Assumption 1 is not satisfied.

The results also indicate that performance achieved by Algorithm 2 is better than that of the CDG, CSF, CDC and MECDA algorithms. Also, the network lifetime achieved by CDG and CSF does not increase when the transmission range of sensor nodes increases. The reason is that, in these two algorithms, all sensor nodes must be constantly activated. Regarding the MECDA algorithm, since it is used to find the routing with the smallest energy consumptions, some sensor nodes are always activated until their energy is depleted. The network is then easier to become disconnected, and thus has smaller lifetime compared to the one achieved by Algorithm 2. Regarding the CDC algorithm, it is based on opportunistic routing. Therefore, the energy of the sensor nodes are more balanced than the case of MECDA. However, it does not guarantee minimum activation of sensor nodes in each



(a) (b) (c)

Fig. 4. Comparison of the network lifetime achieved by different algorithms with: (a) $N = 50$, $M_{\rm cs} = 10$, equal transmission range; (b) $N = 100$, $M_{\rm cs} = 20$, equal transmission range; (c) $N = 100$, $M_{\rm cs} = 20$, unequal transmission range

timeslot. Therefore, it consumes more energy than Algorithm 2 in each timeslot, and has less network lifetime. The average lifetime achieved by Algorithm 2 is significantly longer than the CDG, CSF, CDC and MECDA approaches, which suggests the effectiveness of Algorithm 2 in general scenarios.

Regarding the time complexity, as the complexity of Algorithm 1 is $O(N^2)$ as discussed in our previous paper [10], and recall that this algorithm determines the sensor activation for a single timeslot, the overall time complexity to achieve the approximate network lifetime based on Algorithm 2 is $O(N^2E)$. We further test the computational time to achieve the approximate network lifetime by Algorithm 2 and the upper bound network lifetime by testing the feasibility of Problem (6) with relaxed Constraint (6f). In the settings, $N = 50$ and $M_{\rm cs} = 10$, and the nodes have the same transmission range, which is the same as in Fig. 4 (a). The average computational time to retrieve the network lifetime are 1.4031 seconds and 1.4246 seconds for the normalized transmission range of 0.3 and 0.4, respectively. To calculate the lifetime upper bound $T_{\max}$, one has to test a range between the approximate lifetime T_G and $\sum E_i/M_{\rm cs}$, which requires 4.8974 seconds and 0.7193 seconds for he two aforementioned cases, respectively. The computational time of calculating the lifetime upper bound reduces as the transmission range increases. The reason is that the feasible testing ranges becomes smaller and the approximate lifetime is closer to the upper bound when the node's transmission range increases.

Based on the results above, we conclude that the approach of energy balancing is effective for lifetime maximization.

VI. Conclusion

We considered a dense sensor network for monitoring a one dimensional strip area, such as a pipeline, a tunnel, or a bridge. Given that sensor node replacement is expensive and difficult, the problem of maximizing network lifetime by using compressive sensing was considered. The compressive sensing introduces a cardinality constraint, which makes the problem challenging. Thus, we characterized the performance of an approximation approach based on balancing the residual energy of sensor nodes. We proved that the resulting lifetime is at least 50% of the optimal and that it is near optimal when the ratio of nodal initial energy to nodal energy consumptions is large enough. Simulation results showed that the ratio of the lifetime achieved by balancing the residual energy of the nodes to the upper bound of network lifetime is close to 1 when the WSN is dense enough.

An interesting topic of future work is to study the relationship of energy balancing with lifetime maximization under cardinality constraints in a more general network structure, e.g., a WSN in 2-dimensional free spaces. Besides, deriving solution approaches that apply rounding to the results of the relaxed maximum flow problem is a promising research direction.

Appendix A
Proof of the equivalence of Constraints (1b)-(1d) to Constraint 2

Constraints (1b)-(1d) imply that

$$\begin{aligned} 0 \le E_i(T+1) &= E_i(T) - x_i(T) \\ &= E_i(T-1) - x_i(T-1) - x_i(T) \\ &= \dots \\ &= E_i(1) - \sum_{t=1}^{T} x_i(t) = E_i - \sum_{t=1}^{T} x_i(t)\,, \end{aligned}$$

where the first inequality comes from Constraint (1d), and the equalities come from Constraints (1b) and (1c). Thus, Constraints (1b)-(1d) can be equivalently captured by Constraint 2 ,

Appendix B
Proof of Lemma 1

Proof: We need to show that the solution of Problem (3) can be converted to the solution of Problem (1), and vice versa. Suppose the solution of Problem (3) is $\boldsymbol{z}^* = [z^*(1), \dots, z^*(L)]^T$. Then, we have the solution for Problem (1) to be, $x_i(t) = 1, \forall (t,i) \in \{(t,i) | \sum_{k=1}^{K-1} z^*(k) + 1 \le t \le \sum_{k=1}^{K} z^*(k) \wedge q_K(i) = 1, K = 1, 2, \dots, L\}$, otherwise, $x_i(t) = 0$. That is, activate all the sensor nodes in Profile $\boldsymbol{Q}_1$ for $z^*(1)$ timeslots, and then activate all the sensor nodes in Profile $\boldsymbol{Q}_2$ for $z^*(2)$ timeslots, and so on.

On the other hand, suppose the solution for Problem (1) is $\boldsymbol{x} = \{\boldsymbol{x}(1), \dots, \boldsymbol{x}(T)\}$. Notice that the activated sensor nodes in each timeslot must belong to a feasible activation profile, i.e., $\forall t, \exists l, Q_l = \boldsymbol{x}(t)$. Then, the solution for Problem (3) is

$\boldsymbol{z} = [z(1), \ldots, z(L)]^T$, where $z(i) = \sum_{t=1, Q_i = \boldsymbol{x}(t)}^{T} 1$. This completes the proof. ■

Remark 9: Since the column in matrix $\boldsymbol{Q}$ represents a feasible activation profile, the construction of $\boldsymbol{Q}$ is equivalent to enumerating all the feasible activation profiles. It consists of two steps: 1. Search the route from s_l to s_r; 2. Remove the routes that does not satisfy cardinality constraint. In detail, for the first step, denote $\boldsymbol{s}(v_i, k)$ the set of routes starting from v_i to s_r with k vertices. Then, a dynamic programming based searching could proceed as $\boldsymbol{s}(v_i, k) = \bigcup_{v_j \in \mathcal{N}_-(v_i)} \boldsymbol{s}(v_j, k-1)$. Then, the set of feasible activation profiles could be represented by $\bigcup_{k \geq M+2} \boldsymbol{s}(s_l, k)$. Notice that the number of feasible activation profiles is huge for large and dense networks, and hence this approach cannot be applied in WSNs with limited storage capacity. This approach is just for performance comparison.

Appendix C
Proof of Lemma 2

Proof: The sketch of the proof is based on the one-to-one mapping of the constraints. Constraint (6f) ensues that the flows are unit flows. Together with Constraint (6c), we have that there is at most one unit flow going out from each vertex in each timeslot. Therefore, a unit flow on edge v_i to v_j, $u_{i,j,t} = 1$, represents the activation of both nodes v_i, v_j at timeslot t.

Constraint (6b) represents that the output flow (the first summation) of a node should be equal to the input flow (the second summation) if the node is a sensor node. If the node is v_0 (the sink node s_l), then the difference of its output flow and its input flow should be 1 in each timeslot. If the node is v_{N+1} (the sink node s_r), then the difference should be -1 in each timeslot. Furthermore, Constraints (6c) and (6f) ensure that there is at most one unit flow that goes out from each vertex, i.e, there is one outgoing edge in the trail for each active node. Together with flow conservation Constraint (6b), there is no cyclic in the trail from v_0 to v_{N+1}. This means that the trail is a path. Recall Remark 4 that the connectivity Constraint (1e) is equivalent to the existence of path from v_0 to v_{N+1}. Thus, this flow conservation Constraint (6b) with (6c) and (6f) is equivalent to the connectivity Constraint (1e).

From Constraints (6b), (6c) and (6f), the unit flow represents the activation profile in that timeslot, and $\sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t}$ is either 1 (which means v_i is active) or 0 (which means v_i is inactive) for each sensor node and timeslot. Thus, the summation over time, $\sum_{t=1}^{T} \sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t}$, is the number of timeslots that v_i is activated, and it should be smaller than E_i. Thus, Constraint (6d) is equivalent to the energy budget Constraint (2). Besides, the summation over nodes, $\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}(v_i)} u_{i,j,t}$ represents the number of nodes that is activated in a timeslot, which corresponds to the cardinality constraint. Notice that $v_l \in \mathcal{V}$, and contributes to the summation. On the other hand, even though $v_r \in \mathcal{V}$, $u_{N+1,j,t} = 0$, thus, v_r does not contribute to the summation. Therefore, the right hand side of Constraint (6e) should be $M_{\text{cs}} + 1$.

Given that the objectives of the two problems are identical, and the constraints are equivalent, it is concluded that Problem (1) is equivalent to Problem (6). ■

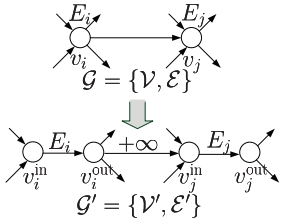


Fig. 5. The transformation of a network with vertex capacity to a network with edge capacity. In the new graph $\mathcal{G}'$, the capacity $C_{ii} = E_i$, $C_{ij} = +\infty$, $C_{jj} = E_j$.

Appendix D
The approach to solve the relaxed Problem (6)

We know that given a fixed T, the relaxed Problem (6) is a linear optimization and thus solvable. Based on this idea, we can use a bisection approach, i.e., we can turn the problem into testing the feasibility of a linear programming problem. The idea is, given a T, we test whether the problem is feasible. If it is feasible, we increase T; otherwise we decrease it, until it converges. Notice that we have already had an upper bound of network lifetime in our previous work [10], which is $\sum E_i / M_{\text{cs}}$, the time complexity of solving the relaxed problem is the time complexity of solving a linear programming problem, which is polynomial, multiplied by $\ln(E_i)$, and it is not high.

Appendix E
Transformation to a maximum flow problem with edge capacities

We need to show Problem (6) can be formulated as a maximum flow problem with edge capacities and cardinality constraints. The transformation follows the standard techniques [38], as shown in Fig. 5. Given a network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, we construct a new directed graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$. The two sink nodes in $\mathcal{V}$, s_l and s_r respectively, are replicated to $\mathcal{V}'$. Every sensor node v_i of initial energy E_i is represented by two nodes v_i^{in} and v_i^{out} connected by a directed arc $\langle v_i^{\text{in}}, v_i^{\text{out}} \rangle$ of capacity E_i in $\mathcal{G}'$. For each edge $\langle v_i, v_j \rangle \in \mathcal{E}$, if $i < j$, we construct a directed arc $\langle v_i^{\text{out}}, v_j^{\text{in}} \rangle$ in $\mathcal{G}'$ with capacity $+\infty$, otherwise we construct a directed arc $\langle v_j^{\text{out}}, v_i^{\text{in}} \rangle$ with capacity $+\infty$ in $\mathcal{G}'$. Thus, the vertex capacity constraints in Problem (6) turn to the edge capacity constraints. For the cardinality constraint, only the edges $\langle v_i^{\text{in}}, v_i^{\text{out}} \rangle$ are taken into accounts. Then, the new maximum flow with edge capacity under cardinality constraint is equivalent to the Problem (6). This completes the proof.

Appendix F
Proof of Lemma 3

Proof: Suppose a route $\boldsymbol{R}_a$ from s_l to s_r contains no backward arcs, and can be divided into three parts $\boldsymbol{R}_a^1 = \langle s_l, \ldots, v_1' \rangle$, $\boldsymbol{R}_a^2 = \langle v_1', \ldots, v_2' \rangle$, $\boldsymbol{R}_a^3 = \langle v_2', \ldots, s_r \rangle$. The sensor nodes in $\boldsymbol{R}_a$ are activated at t_1. The lifetime of an activation is $t_2 - 1$, i.e., at time $t_2 > t_1$, we cannot find any

unblocked routes from s_l to s_r that contains no backward arcs. However, if we can find a route $\boldsymbol{R}_b$ that contains at least one backward arc, $\boldsymbol{R}_b$ can also be divided into three parts $\boldsymbol{R}_b^1 = \langle s_l, \ldots, v_2' \rangle$, $\boldsymbol{R}_b^2 = \langle v_2', \ldots, v_1' \rangle$ and $\boldsymbol{R}_b^3 = \langle v_1', \ldots, s_r \rangle$, where $\boldsymbol{R}_b^1$ and $\boldsymbol{R}_b^3$ contain no backward arcs. If both $\langle \boldsymbol{R}_a^1, \boldsymbol{R}_b^3 \rangle$ and $\langle \boldsymbol{R}_b^1, \boldsymbol{R}_a^3 \rangle$ satisfy the cardinality constraint, we can pick route $\langle \boldsymbol{R}_a^1, \boldsymbol{R}_b^3 \rangle$ at t_1 and $\langle \boldsymbol{R}_b^1, \boldsymbol{R}_a^3 \rangle$ at t_2, such that the WSN lifetime increases from $t_2 - 1$ to t_2, which completes the proof. ∎

Appendix G
Proof of Lemma 4

Proof: The proof is by contradiction. Suppose that a backward arc $\langle v_i^{\text{out}}, v_i^{\text{in}} \rangle$ exists in $\boldsymbol{R}_b$, we divide $\boldsymbol{R}_b$ into three parts, $\boldsymbol{R}_b^1$, $\boldsymbol{R}_b^2$, $\boldsymbol{R}_b^3$, where $\boldsymbol{R}_b^1 = \langle s_l, \ldots, v_u^{\text{out}}, v_j^{\text{in}} \rangle$, $\boldsymbol{R}_b^2 = \langle v_j^{\text{in}}, \ldots, v_i^{\text{out}}, v_i^{\text{in}}, \ldots, v_k^{\text{out}} \rangle$, $\boldsymbol{R}_b^3 = \langle v_k^{\text{out}}, v_w^{\text{in}}, \ldots, s_r \rangle$, such that backward arcs only exist in $\boldsymbol{R}_b^2$. As $\boldsymbol{R}_b$ is unblocked, $\boldsymbol{R}_b^2$ is unblocked. If $\boldsymbol{R}_b$ leads to suboptimal network lifetime, there is a route $\boldsymbol{R}_a = \langle s_l, \ldots, v_k^{\text{in}}, v_k^{\text{out}}, \ldots, v_i^{\text{in}}, v_i^{\text{out}}, \ldots, v_j^{\text{in}}, v_j^{\text{out}}, \ldots, s_r \rangle$ was chosen to be activated at a time $t_2 \leq t_1$, where $\langle v_k^{\text{out}}, \ldots, v_i^{\text{in}}, v_i^{\text{out}}, \ldots, v_j^{\text{in}} \rangle$ is the inverse sequence of $\boldsymbol{R}_b^2$. Similarly, we divide R_a into three parts, $\boldsymbol{R}_a^1 = \langle s_l, \ldots, v_k^{\text{out}} \rangle$, $\boldsymbol{R}_a^2 = \langle v_k^{\text{out}}, \ldots, v_j^{\text{in}} \rangle$, and $\boldsymbol{R}_a^3 = \langle v_j^{\text{out}}, \ldots, s_r \rangle$, where $\boldsymbol{R}_a^2$. Then we have the maximum flow increment of $\boldsymbol{R}_a^1$ and $\boldsymbol{R}_a^3$ at t_2 satisfies $\delta_a^1(t_2) \geq 1$, $\delta_a^3(t_2) \geq 1$, and the maximum flow increment of $\boldsymbol{R}_b^1$ and $\boldsymbol{R}_b^3$ at t_2 satisfies $\delta_b^1(t_2) \geq \delta_b^1(t_1+1) \geq 1$, $\delta_b^3(t_2) \geq \delta_b^3(t_1+1) \geq 1$. (Otherwise, $\boldsymbol{R}_a$ is blocked at t_2 and $\boldsymbol{R}_b$ is blocked at $t_1 + 1$.)

Notice that route $\langle \boldsymbol{R}_a^1, \boldsymbol{R}_b^3 \rangle$ and route $\langle \boldsymbol{R}_b^1, \boldsymbol{R}_a^3 \rangle$ are not blocked at t_2, and they contain no backward arcs. Then, according to Line 4 to Line 5 in Algorithm 1, which minimize the number of nodes to be activated in each time slot, we have that $|\boldsymbol{R}_a^1|$, $|\boldsymbol{R}_a^3|$, $|\boldsymbol{R}_b^1|$ and $|\boldsymbol{R}_b^3|$, the number of sensor nodes in $\boldsymbol{R}_a^1$, $\boldsymbol{R}_a^3$, $\boldsymbol{R}_b^1$ and $\boldsymbol{R}_b^3$ respectively, should satisfy $|\boldsymbol{R}_b^3| > |\boldsymbol{R}_a^3|$, $|\boldsymbol{R}_b^1| > |\boldsymbol{R}_a^1|$. Thus, the sensor node v_w lies between v_i and v_j, and the node v_u lies between v_k and v_i. From Assumption 1, we have $v_u \in \mathcal{N}_-(v_k)$. Further, as arc $\langle v_k^{\text{out}}, v_w^{\text{in}} \rangle$ is in $\boldsymbol{R}_b^3$ $v_w \in \mathcal{N}_-(v_k)$, and hence v_u and v_w is connected.

Moreover, as $\boldsymbol{R}_b$ leads to the suboptimality of the network lifetime, we have that route $\langle \boldsymbol{R}_b^1, \boldsymbol{R}_a^3 \rangle$ and route $\langle \boldsymbol{R}_a^1, \boldsymbol{R}_b^3 \rangle$ satisfy cardinality constraint, i.e., $|\boldsymbol{R}_b^1| + |\boldsymbol{R}_a^3| \geq M_{\text{cs}}$ and $|\boldsymbol{R}_a^1| + |\boldsymbol{R}_b^3| \geq M_{\text{cs}}$. Together with $|\boldsymbol{R}_b^1| > |\boldsymbol{R}_a^1|$, $|\boldsymbol{R}_b^3| > |\boldsymbol{R}_a^3|$, we have that $|\boldsymbol{R}_b^1| + |\boldsymbol{R}_b^3| \geq M_{\text{cs}}$.

Then we have an unblocked route that satisfies cardinality constraint and has no backward arcs, $\langle \boldsymbol{R}_b^{1'}, \boldsymbol{R}_b^{3'} \rangle$, at $t_1 + 1$ where $\boldsymbol{R}_b^{1'}$ is the route $\boldsymbol{R}_b^1$ without v_j^{in}, and $\boldsymbol{R}_b^{3'}$ is the route $\boldsymbol{R}_b^3$ without v_k^{out}. This comes in contradiction with that no unblocked route with only forward arcs can be found at t_1+1. Thus, $\boldsymbol{R}_b$ does not contain any backward arc $\langle v_i^{\text{out}}, v_i^{\text{in}} \rangle$ for all i, which completes the proof. ∎

Appendix H
Proof of Lemma 5

We first need a lemma that is used in the proof for Lemma 5:

Lemma 6: Consider two positive integers E_i and E_j that satisfy $E_i < E_j$. For any positive integers x and y, if x and y satisfy $(y-1)/E_j < x/E_i \leq y/E_j$, then we have $(x-1)/E_i < (y-1)/E_j$.

Proof: It suffices to show that $(x-1)/E_i < (y-1)/E_j$. Since $x/E_i \leq y/E_j$, we have that

$$\frac{x-1}{E_i} < \frac{x}{E_i} - \frac{1}{E_j} \leq \frac{y}{E_j} - \frac{1}{E_j} = \frac{y-1}{E_j},$$

where the first inequality comes from $E_i < E_j$. This concludes the proof. ∎

Then, Lemma 5 is proved as follows:

Proof of Lemma 5: Suppose that backward arc $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$ with $j > i$ exists in an unblocked route $\boldsymbol{R}_b$ when the network expires at $t = t_1$. Then there exists a node v_h^{out} and v_k^{in} in $\boldsymbol{R}_b$ such that route $\langle v_h^{\text{out}}, v_j^{\text{in}}, v_i^{\text{out}}, v_k^{\text{in}} \rangle$ is in $\boldsymbol{R}_b$ and $h < i < j < k$. According to Assumption 1 and 2, forward arcs $\langle v_h^{\text{out}}, v_i^{\text{in}} \rangle$ and $\langle v_j^{\text{out}}, v_k^{\text{in}} \rangle$ exist in $\mathcal{G}'$. As $\boldsymbol{R}_b$ leads to the suboptimality in network lifetime, we have that $v_i \in \mathcal{N}_-(v_x)$ for any node v_x that $v_h \in \mathcal{N}_-(v_x)$, that $v_y \in \mathcal{N}_-(v_j)$ for any node $v_y \in \mathcal{N}_-(v_k)$, and that there is no direct edge between v_h^{out} and v_k^{in}, as shown in Fig. 2.

Then we focus on the part in $\boldsymbol{R}_b$ that contains backward arcs. Similar to the proof for Lemma 4, we again divide $\boldsymbol{R}_b$ into three parts, $\boldsymbol{R}_b^1 = \langle s_l, \ldots, v_h^{\text{in}}, v_h^{\text{out}} \rangle$, $\boldsymbol{R}_b^2 = \langle v_h^{\text{out}}, v_j^{\text{in}}, v_i^{\text{out}}, v_k^{\text{in}} \rangle$, and $\boldsymbol{R}_b^3 = \langle v_k^{\text{in}}, v_k^{\text{out}}, \ldots, s_r \rangle$. If $\boldsymbol{R}_b$ causes the suboptimality in network lifetime, we have that the maximum flow increment of $\boldsymbol{R}_b^1$, $\boldsymbol{R}_b^2$, $\boldsymbol{R}_b^3$, should satisfy $\delta_b^1(t_1) \geq 1$, $\delta_b^2(t_1) \geq 1$, and $\delta_b^3(t_1) \geq 1$. Let $E_i(t)$ be the residual energy of sensor node v_i before the activation at t-th slot, $E_i(0) = E_i$ be the initial energy of sensor node v_i. As $\delta_b^2(t_1) \geq 1$, we have that a route that contains v_i and v_j were chosen for activation at $t_2 < t_1$, which means that $E_i(t_2)/E_i + E_j(t_2)/E_j \geq E_h(t_2)/E_h + E_j(t_2)/E_j$ and $E_i(t_2)/E_i + E_j(t_2)/E_j \geq E_i(t_2)/E_i + E_k(t_2)/E_k$ at t_2 according to Algorithm 1, which chooses the nodes having the maximum sum of normalized residual energy in every timeslot.

Then, we divide the analysis into four cases: 1) $E_h < E_i$ and $E_k < E_j$; 2) $E_h < E_i$ and $E_k \geq E_j$; 3) $E_h \geq E_i$ and $E_k < E_j$; 4) $E_h \geq E_i$ and $E_k \geq E_j$. We will show that for case 1), there will be no flow increment; for case 2), 3) and 4), there will be at most 1.

In case 1), $E_h < E_i$ and $E_k < E_j$. In the initial time when none of these four nodes have been activated, then $E_h(0)/E_h = E_i(0)/E_i = E_j(0)/E_j = E_k(0)/E_k = 1$. According to Lemma 6, we have $0/E_h < 1/E_i < 1/E_h$. It means that sensor node v_h expires earlier than sensor node v_i, as the sum of residual energy of $\langle v_h, v_j \rangle$, which is $1/E_h + E_j(t)/E_j$, is always larger than that of $\langle v_i, v_j \rangle$, which is $1/E_i + E_j(t)/E_j$. Similarly, we have $0/E_k < 1/E_j < 1/E_k$ and hence sensor node v_k expires earlier than sensor node v_j. Once either v_h or v_k expires, the route R_b is blocked even when the network has not expired, which contradicts that R_b is unblocked when the network expires. Thus, there will be no flow increment in this case.

In case 2), $E_h < E_i$ and $E_k \geq E_j$. If $E_h \leq E_j$, with similar reason to case 1), we have that v_h expires first, and then R_b is blocked, which is in contradiction with that R_b is unblocked when the network expires. It means that $E_h > E_j$ as we can find an unblocked route R_b when the network expires. Thus,

v_j expires earlier than v_h. If $E_j \le E_k \le E_i$, we have that after v_j expires, $\langle v_h, v_j\rangle$ and $\langle v_i, v_j\rangle$ is blocked. The algorithm will pick $\langle v_i, v_k\rangle$ until v_k expires. Then R_b is blocked as $\langle v_k^{\text{in}}, v_k^{\text{out}}\rangle$ is blocked. Consequently, $E_k > E_i > E_h > E_j$. Then, in the initial time when none of these four sensor nodes have been activated, the normalized residual energy of these four nodes is 1. Consequently, once $\langle v_i, v_j\rangle$ is chosen, suppose at t_2 when all these four nodes are not activated before, we have

$$\frac{E_i(t_2)}{E_i(0)} = \frac{E_j(t_2)}{E_j(0)} = \frac{E_k(t_2)}{E_k(0)} = \frac{E_h(t_2)}{E_h(0)} = 1\,,$$

$$\frac{E_k(t_2+1)}{E_k(0)} > \frac{E_j(t_2+1)}{E_j(0)}, \qquad \frac{E_h(t_2+1)}{E_h(0)} > \frac{E_i(t_2+1)}{E_i(0)}\,,$$

It directly gives us that, in the next timeslot t_2+1,

$$\frac{E_k(t_2+1)}{E_k} + \frac{E_i(t_2+1)}{E_i} > \frac{E_j(t_2+1)}{E_j} + \frac{E_i(t_2+1)}{E_i} \tag{7}$$

Also, we have that

$$\begin{aligned}
&\frac{E_k(t_2+1)}{E_k(0)} + \frac{E_i(t_2+1)}{E_i(0)} \\
&= \frac{E_k(t_2)}{E_k(0)} + \frac{E_i(t_2)-1}{E_i(0)} &(8)\\
&= 2 - \frac{1}{E_i(0)} > 2 - \frac{1}{E_j(0)} &(9)\\
&= \frac{E_h(t_2)}{E_h(0)} + \frac{E_j(t_2)-1}{E_j(0)} = \frac{E_h(t_2+1)}{E_h(0)} + \frac{E_j(t_2+1)}{E_j(0)}\,, &(10)
\end{aligned}$$

where (8) holds since v_k is not activated but v_i is activated at t_2, (9) holds since $E_i > E_j > 0$, and (10) holds since v_h is not activated but v_j is activated at t_2.

Then the algorithm will choose $\langle v_i, v_k\rangle$ instead of $\langle v_i, v_j\rangle$ and $\langle v_h, v_j\rangle$, due to (7) and (10). After this activation, as $E_k > E_j$, we have $(E_i(t_2+1)-2)/E_i + (E_k(t_2+1)-1)/E_k > (E_i(t_2+1)-2)/E_i + (E_j(t_2+1)-1)/E_j$. It means that the normalized residual energy of route $\langle v_i, v_j\rangle$ is still smaller than that of $\langle v_i, v_k\rangle$ and $\langle v_h, v_j\rangle$. The algorithm will then pick $\langle v_i, v_k\rangle$ until the normalized residual energy of v_k is smaller than that of v_j. When this happens, we have that the normalized residual energy of v_i is smaller than that of v_k, and hence smaller than v_j and v_h. It means that the algorithm will then pick $\langle v_h, v_j\rangle$. After that, the normalized residual energy of $\langle v_i, v_k\rangle$ is again larger than that of $\langle v_i, v_j\rangle$ according to Lemma 6. This indicates that the algorithm will always pick $\langle v_i, v_k\rangle$ or $\langle v_h, v_j\rangle$ instead of $\langle v_i, v_j\rangle$. Consequently, the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}}\rangle$ in this case is 1 as $\langle v_i, v_j\rangle$ was chosen only once.

As case 3) is symmetric to case 2), we have the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}}\rangle$ in this case is also 1.

In case 4), $E_h \ge E_i$ and $E_k \ge E_j$. We have for any positive integer x, $x/E_i \ge x/E_h$, and $x/E_j \ge x/E_k$. In the initial time when none of these four sensor nodes have been activated, the normalized residual energy of these four nodes is equal to 1. Hence, once $\langle v_i, v_j\rangle$ is chosen, suppose at t_2, we have

$$\frac{E_h(t_2+1)}{E_h(0)} > \frac{E_i(t_2+1)}{E_i(0)}, \tag{11}$$

$$\frac{E_k(t_2+1)}{E_k(0)} > \frac{E_j(t_2+1)}{E_j(0)}\,, \tag{12}$$

and thus

$$\begin{aligned}
\frac{E_i(t_2+1)}{E_i(0)} + \frac{E_j(t_2+1)}{E_j(t_2+1)} &< \frac{E_i(t_2+1)}{E_i(0)} + \frac{E_k(t_2+1)}{E_k(0)}\,,\\
\frac{E_i(t_2+1)}{E_i(0)} + \frac{E_j(t_2+1)}{E_j(t_2+1)} &< \frac{E_h(t_2+1)}{E_h(0)} + \frac{E_j(t_2+1)}{E_j(0)}\,.
\end{aligned}$$

Thus, the algorithm will choose $\langle v_i, v_k\rangle$ or $\langle v_h, v_j\rangle$ instead of $\langle v_i, v_j\rangle$. Due to the symmetry, the analysis when it picks $\langle v_i, v_k\rangle$ is similar to that if it picks $\langle v_h, v_j\rangle$. Thus, we only analyze the case if it picks $\langle v_i, v_k\rangle$.

Suppose that the algorithm picks $\langle v_i, v_k\rangle$, then the normalized residual energy of sensor node v_i and v_k becomes $(E_i(t_2+1)-1)/E_i$ and $(E_k(t_2+1)-1)/E_k$. Suppose $(E_k(t_2+1)-1)/E_k > E_j(t_2+1)/E_j$, we have that the normalized residual energy of route $\langle v_i, v_j\rangle$ is still less than that of $\langle v_i, v_k\rangle$ and $\langle v_h, v_j\rangle$. Otherwise, we have

$$\begin{aligned}
&\frac{E_j(t_2+1)-1}{E_j} + \frac{E_h(t_2+1)}{E_h}\\
&> \frac{E_j(t_2+1)-1}{E_j} + \frac{E_i(t_2+1)}{E_i}\\
&> \frac{E_j(t_2+1)-1}{E_j} + \frac{E_i(t_2+1)-1}{E_i}\,,
\end{aligned}$$

where the first inequality comes from (11). The summation of residual energy of v_j and v_h is larger than that of v_i and v_j. Thus, the algorithm will pick route $\langle v_h, v_j\rangle$. Then the normalized residual energy of these four sensor nodes are $(E_h(t_2+1)-1)/E_h$, $(E_i(t_2+1)-1)/E_i$, $(E_j(t_2+1)-1)/E_j$ and $(E_k(t_2+1)-1)/E_k$. As $1/E_i > 1/E_h$, $1/E_j > 1/E_k$, we have

$$\begin{aligned}
&\frac{E_i(t_2+1)-1}{E_i(0)} + \frac{E_j(t_2+1)-1}{E_j(0)}\\
&= \frac{E_i(t_2+1)}{E_i(0)} - \frac{1}{E_i(0)} + \frac{E_j(t_2+1)-1}{E_j(0)}\\
&< \frac{E_h(t_2+1)}{E_i(0)} - \frac{1}{E_h(0)} + \frac{E_j(t_2+1)-1}{E_j(0)}\\
&= \frac{E_h(t_2+1)-1}{E_h(0)} + \frac{E_j(t_2+1)-1}{E_j}\,,
\end{aligned} \tag{13}$$

and similarly

$$\begin{aligned}
&\frac{E_i(t_2+1)-1}{E_i(0)} + \frac{E_j(t_2+1)-1}{E_j(0)}\\
&< \frac{E_i(t_2+1)-1}{E_i(0)} + \frac{E_k(t_2+1)-1}{E_k(0)}\,.
\end{aligned} \tag{14}$$

(13) and (14) indicate that the algorithm will always pick $\langle v_i, v_k\rangle$ or $\langle v_h, v_j\rangle$ instead of $\langle v_i, v_j\rangle$, until v_i or v_j expires. Consequently, the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}}\rangle$ in this case is 1, as $\langle v_i, v_j\rangle$ was only picked once.

To sum up, the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$ is 1 for Cases 2) to 4), and it is 0 for Case 1). Thus, the maximum flow increment of $\langle v_j^{\text{in}}, v_i^{\text{out}} \rangle$ is 1. ■

REFERENCES

[1] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–39, 2009.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. IEEE international conference on System sciences*, 2000, pp. 1–10.

[3] V.-T. Hoang, N. Julien, and P. Berruet, "Cluster-head selection algorithm to enhance energy-efficiency and reliability of wireless sensor networks," in *Proc. European Wireless Conference*, 2014, pp. 1–6.

[4] J.-S. Leu, T.-H. Chiang, M.-C. Yu, and K.-W. Su, "Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes," *IEEE Communications Letters*, vol. 19, no. 2, pp. 259–262, 2015.

[5] Y. Chen and A. Krause, "Near-optimal batch mode active learning and adaptive submodular optimization," in *Proc. International Conference on Machine Learning*, 2013, pp. 160–168.

[6] S. K. A. Imon, A. Khan, M. Di Francesco, and S. K. Das, "Rasmalai: A randomized switching algorithm for maximizing lifetime in tree-based wireless sensor networks," in *Proc. IEEE Conference on Computer Communications*, 2013, pp. 2913–2921.

[7] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.

[8] G. P. Halkes, T. van Dam, and K. Langendoen, "Comparing energy-saving mac protocols for wireless sensor networks," *Mobile Networks and Applications*, vol. 10, no. 5, pp. 783–791, 2005.

[9] G. Degirmenci, J. P. Kharoufeh, and O. A. Prokopyev, "Maximizing the lifetime of query-based wireless sensor networks," *ACM Trans. Sensor Networks (TOSN)*, vol. 10, no. 4, pp. 56:1–56:24, 2014.

[10] R. Du, L. Gkatzikis, C. Fischione, and M. Xiao, "Energy efficient sensor activation for water distribution network based on compressive sensing," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2997–3010, 2015.

[11] H. S. AbdelSalam and S. Olariu, "Toward adaptive sleep schedules for balancing energy consumption in wireless sensor networks," *IEEE Trans. Computers*, vol. 61, no. 10, pp. 1443–1458, 2012.

[12] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[13] L. Xu, X. Qi, Y. Wang, and T. Moscibroda, "Efficient data gathering using compressed sparse functions," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2013, pp. 310–314.

[14] R. Du, L. Gkatzikis, C. Fischione, and M. Xiao, "Energy efficient monitoring of water distribution networks via compressive sensing," in *Proc. IEEE International Conference on Communications (ICC)*, 2015, pp. 8309–8314.

[15] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Trans. Networking (TON)*, vol. 12, no. 4, pp. 609–619, 2004.

[16] A. Jarry, P. Leone, O. Powell, and J. Rolim, "An optimal data propagation algorithm for maximizing the lifespan of sensor networks," in *Distributed Computing in Sensor Systems*. Springer, 2006, pp. 405–421.

[17] C. Cassandras, T. Wang, and S. Pourazarm, "Optimal routing and energy allocation for lifetime maximization of wireless sensor networks with nonideal batteries," *IEEE Trans. Control of Network Systems*, vol. 1, no. 1, pp. 86–98, March 2014.

[18] T. Shi, S. Cheng, Z. Cai, and J. Li, "Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks," in *Proc. IEEE International Conference on Computer Communications*, 2016, pp. 1566–1574.

[19] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "Atpc: adaptive transmission power control for wireless sensor networks," in *Proc. ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 223–236.

[20] Z. Fan, S. Bai, S. Wang, and T. He, "Delay-bounded transmission power control for low-duty-cycle sensor networks," *IEEE Trans. Wireless Communications,*, vol. 14, no. 6, pp. 3157–3170, 2015.

[21] D. Incebacak, R. Zilan, B. Tavli, J. M. Barcelo-Ordinas, and J. Garcia-Vidal, "Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode," *Ad Hoc Networks*, vol. 24, pp. 134–147, 2015.

[22] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors Journal*.

[23] A. Cammarano, D. Spenza, and C. Petrioli, "Energy-harvesting wsns for structural health monitoring of underground train tunnels," in *Proc. IEEE Conference on Computer Communications Workshops*, 2013, pp. 75–76.

[24] A. Berger, L. B. Hormann, C. Leitner, S. B. Oswald, P. Priller, and A. Springer, "Sustainable energy harvesting for robust wireless sensor networks in industrial applications," in *Proc. IEEE Sensors Applications Symposium*. IEEE, 2015, pp. 1–6.

[25] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff, "On renewable sensor networks with wireless energy transfer: the multi-node case," in *Proc. Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012, pp. 10–18.

[26] R. Du, C. Fischione, and M. Xiao, "Lifetime maximization for sensor networks with wireless energy transfer," in *Proc. IEEE International Conference on Communications (ICC) (to appear)*, 2016.

[27] H. Zhang and H. Shen, "Balancing energy consumption to maximize network lifetime in data-gathering sensor networks," *IEEE Trans. Parallel and Distributed Systems (TPDS)*, vol. 20, no. 10, pp. 1526–1539, 2009.

[28] F. Ren, J. Zhang, T. He, C. Lin, and S. K. Ren, "Ebrp: energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE Trans. Parallel and Distributed Systems (TPDS)*, vol. 22, no. 12, pp. 2108–2125, 2011.

[29] R. Misra and C. Mandal, "Rotation of cds via connected domatic partition in ad hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 8, no. 4, pp. 488–499, 2009.

[30] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2009, pp. 145–156.

[31] J. Kleinberg and E. Tardos, *Algorithm design*. Pearson Education India, 2006.

[32] D. Bertsimas and R. Demir, "An approximate dynamic programming approach to multidimensional knapsack problems," *Management Science*, vol. 48, no. 4, pp. 550–565, 2002.

[33] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithm, and applications*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[34] G. Como, E. Lovisari, and K. Savla, "Throughput optimality and overload behavior of dynamical flow networks under monotone distributed routing," *IEEE Trans. Control of Network Systems*, vol. 2, no. 1, pp. 57–67, 2015.

[35] C. Luo, J. Sun, and F. Wu, "Compressive network coding for approximate sensor data gathering," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–6.

[36] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. Vasilakos, and M.-Y. Wu, "Cdc: Compressive data collection for wireless sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188 – 2197, 2014.

[37] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *Proc. IEEE conference on Sensor, mesh and ad hoc communications and networks (SECON)*, 2011, pp. 46–54.

[38] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 2013.